

19^o Treino para alunos da UFPR

5 de Maio de 2017

Sevidor BOCA:
<http://maratona.c3sl.ufpr.br/>



Flávio Zavan
Ricardo Oliveira

Instruções Importantes

- Em cada problema, cada arquivo de entrada contém apenas um caso de teste. Sua solução será executada com vários arquivos de entrada.
- Se a solução der erro ou esgotar o tempo limite para um dado arquivo de entrada, você receberá a indicação de erro (estouro de tempo, resposta errada, etc.) para aquele arquivo, e a execução terminará. O arquivo que causou o erro não é identificado. Note que pode haver outros erros, de outros tipos, para outros arquivos de entrada, mas apenas o primeiro erro encontrado é reportado.
- Sua solução será compilada com a seguinte linha de comando:
 - C: `gcc -static -O2 -lm`
 - C++: `g++ -static -O2 -lm`
 - C++11: `g++ -std=c++11 -static -O2 -lm`
 - Java: `javac`
 - Pascal: `fpc -Xt -XS -O2`
 - Python: `python3`
- Sua solução deve processar cada arquivo de entrada no tempo máximo estipulado para cada problema, dado pela seguinte tabela:

Problema	Nome	Tempo Limite (segundos)
A	Ascensão Acumulada	1
B	AAAAA	1
C	Terceira Guerra	1
D	Forever Alone	1
E	Briga de Galo	1

- Os juizes usam um sistema de 64 bits (idêntico às máquinas do DINF).
- Todas as linhas, tanto na entrada quanto na saída, terminam com o caractere de fim-de-linha ($\backslash n$), mesmo quando houver apenas uma única linha no arquivo.
- Para submissões em **JAVA**, a classe deverá ter o mesmo nome que o *basename* do problema (leia a linha entre o título e o texto do problema).

A: Ascensão Acumulada

Arquivo: `ascensao.[c|cpp|java|pas|py]`

Vasya, Petya e seus amigos são montanhistas de fim de semana e atualmente estão realizando uma competição. Todo fim de semana, cada um realiza um roteiro em montanhas e aquele que tiver a maior ascensão acumulada, vence.

Como sempre chegam muito cansados em casa após um longo dia de caminhada, nunca conseguem computar o vencedor. Por este motivo pediram que você escreva um programa que faça justamente isto para eles.

Entrada

A primeira linha de entrada consiste em dois números inteiros L , o número de locais conhecidos, e N , o número de amigos, ($1 \leq L, N \leq 100$). As L linhas seguintes descrevem os locais conhecidos, cada um descrito por uma linha: primeiro o nome do local em letras maiúsculas e minúsculas, com até 100 caracteres; seguida de um número inteiro entre 0 e 10000, indicando a altitude em metros naquele ponto.

Em seguida são descritos os roteiros dos N amigos, cada um iniciado por uma linha contendo o nome do amigo (apenas letras, sem espaços, até 32 caracteres) e o número de locais visitados em seu roteiro (de 1 a 100). Os locais são indicados em ordem, pelo seu nome e cada um em uma linha. Cada amigo tem um nome diferente e é garantido que todos os locais visitados foram listados anteriormente.

Saída

Deve-se imprimir exatamente uma linha contendo o nome do amigo que fez o roteiro com a maior ascensão acumulada e o valor total, em metros, de ascensão realizada no roteiro. A resposta sempre será única, isto é, nunca haverá dois amigos vencedores.

No exemplo da próxima página, quem vence é Petya. Embora Vasya tenha subido ao ponto mais alto, ele acumulou apenas 935 metros de ascensão, enquanto Petya começou de uma altitude baixa e conseguiu acumular 1947 metros. Ricardo acumulou apenas 496 metros e ficou em último

Exemplo de entrada	Exemplo de saída
<p>10 3 MorroDoGetulio 1482 BairroAltoDeAntonina 110 CruzoCaratuvaPP 1485 DepressaoEntreCaratuvaPP 1463 FazendaRioDasPedras 986 PontesobreRioCotia 589 PicoParana 1877 PicoFerraria 1745 BrejoComQuicaca 1415 Taipabucu 1727 Vasya 9 FazendaRioDasPedras MorroDoGetulio CruzoCaratuvaPP DepressaoEntreCaratuvaPP PicoParana DepressaoEntreCaratuvaPP CruzoCaratuvaPP MorroDoGetulio FazendaRioDasPedras Petya 8 BairroAltoDeAntonina PontesobreRioCotia PicoFerraria BrejoComQuicaca Taipabucu CruzoCaratuvaPP MorroDoGetulio FazendaRioDasPedras Ricardo 3 FazendaRioDasPedras MorroDoGetulio FazendaRioDasPedras</p>	<p>Petya 1947</p>

B: AAAAA

Arquivo: AAAAA.[c|cpp|java|pas|py]

AAAAAAAAA A A AAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAA AAAAAAAAAA-
 AAAAAAAAAAAAAAAAAA AAAAAAAAAA AAAAAAAAAAAAAAAAAA AAAAAAAAAA-
 AAAA, AAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAA AAAAAAAAAA AAAAA AAAA
 AA AA AAAAA AA AAAAAA.

AAAA, AAAA AAAAAAAAAA! AAAAA AAAAAAAAAA A AAAAAA AAA AA AAAAA-
 AAA A AAAAA AAAAAA. AAAA AAAAA AA A AAAAA... AAAAAA AAA AAAAA-
 AAAAA AAAAA AAAAAA AAAAA AAAAA A A AAAAA!

AAAAAAAAAAAAAAAAA A A AAAAAAAAAAAAAAAAAA A AAAAAA AAA AAAAAAAAAA AAAA
 AA AAAAA AAAAAAAAAA AAAAAAAAAA A AAA AAAAAA A A A AAAAAAAAAAAAA-
 AAAAAAAAAA AAAA AAAAA AAA AAA AA AAAAAA.

Entrada

AAAAAAA AAAA AAAAA AAA AAAAA A ($1 \leq A \leq 100$), AA AA AAAAAAAAA
 AAA AAAA.

Saída

AAAAAAA A AAAAA A A AAAAA¹.

Exemplo de entrada	Exemplo de saída
1	A

Exemplo de entrada	Exemplo de saída
2	AA

Exemplo de entrada	Exemplo de saída
3	AAA

¹Atenção para a penúltima instrução importante que consta na página 2 desta prova!

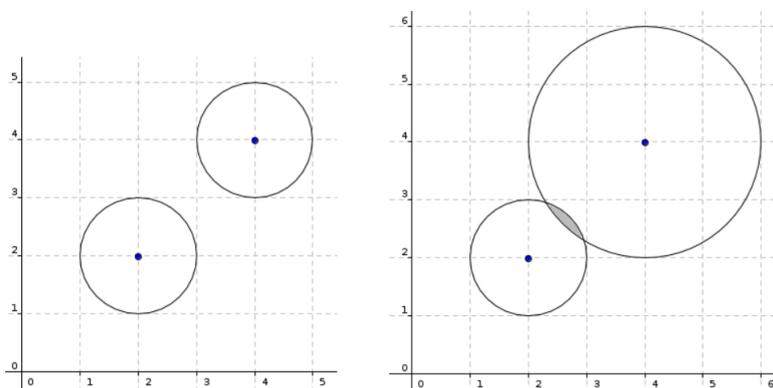
C: Terceira Guerra

Arquivo: terceira.[c|cpp|java|pas|py]

O clima internacional está tenso. As poderosas nações da Nlogônia e da Quadridônia têm testado bombas ultimamente. Por isso, as pessoas temem que elas iniciem a 3ª guerra mundial.

As duas nações testarão uma nova bomba cada, simultaneamente! Cada bomba tem seu *ponto de impacto* e seu *raio de alcance*, o que define seu *círculo de cobertura*. Se uma bomba interferir no teste da outra, a guerra irá começar! Em outras palavras, a guerra começará se a área da intersecção dos círculos de cobertura das duas bombas for *estritamente positiva*.

As figuras abaixo mostram os exemplos de entrada dados. No primeiro caso, não há intersecção dos círculos e, portanto, não haverá guerra. No segundo caso, a área de intersecção dos círculos é positiva (indicada pela área em cinza) e, portanto, haverá guerra.



Entrada

A primeira linha contém três inteiros X_1 , Y_1 e R_1 ($-100 \leq X_1, Y_1 \leq 100, 1 \leq R_1 \leq 100$), indicando as coordenadas do ponto de impacto e o raio de alcance da bomba da Nlogônia. A segunda linha contém três inteiros X_2 , Y_2 e R_2 ($-100 \leq X_2, Y_2 \leq 100, 1 \leq R_2 \leq 100$), indicando as coordenadas do ponto de impacto e o raio de alcance da bomba da Quadridônia.

Saída

Imprima uma linha contendo **guerra** se haverá guerra, ou **paz** caso contrário.

Exemplo de entrada	Exemplo de saída
2 2 1 4 4 1	paz

Exemplo de entrada	Exemplo de saída
2 2 1 4 4 2	guerra

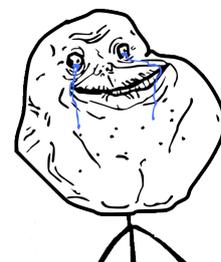
D: Forever Alone

Arquivo: `forever.[c|cpp|java|pas|py]`

A *Googlbok* acabou de disponibilizar sua nova rede social, o *FacePlus*, à população de Nlogônia. Já no primeiro dia de uso da rede, a empresa percebeu que as amizades em Nlogônia se formavam de maneira muito peculiar.

Existem N pessoas em Nlogônia, numeradas de 1 a N . Para cada par de pessoas A e B , elas são amigas entre si se e somente se A divide B ou vice-versa. Por exemplo, considere que $N = 30$. As pessoas 4 e 12 são amigas (pois 4 divide 12), e as pessoas 12 e 24 também são (pois 12 divide 24). De fato, se $N = 30$, então as pessoas que são amigas da pessoa 12 são: 1, 2, 3, 4, 6, 12 e 24. Note que toda amizade é mútua, e que toda pessoa é considerada amiga dela mesma.

A pessoa que tem o *menor* número de amigos em Nlogônia é chamada de *Forever Alone*. Note que mais de uma pessoa pode ser considerada *Forever Alone* se há mais de uma pessoa com o menor número de amigos. Sua tarefa é, dados N e uma pessoa P , determinar se P é uma *Forever Alone*.



Meme ultrapassado mas que você ainda lembra

Entrada

A entrada contém uma linha com dois inteiros N e P ($4 \leq N \leq 2 \times 10^9, 1 \leq P \leq N$), o número de pessoas em Nlogônia e a pessoa dada, respectivamente.²

Saída

Imprima uma linha contendo `sim` se a pessoa dada P é *Forever Alone*, ou `nao` caso contrário.

Exemplo de entrada	Exemplo de saída
30 12	nao

Exemplo de entrada	Exemplo de saída
30 13	nao

Exemplo de entrada	Exemplo de saída
30 17	sim

²Dica para iniciantes: Em Pascal, o tipo *integer* tem apenas 16 bits, enquanto o tipo *longint* tem 32 bits.

E: Briga de Galo

Arquivo: galo.[c|cpp|java|pas|py]

Todo ano bissexto é organizado o campeonato mundial de briga de galo, esporte no qual duplas competem dentro de uma piscina com o objetivo de derrubar os oponentes ³. Entretanto, como todo esporte organizado, existem várias regras definindo o que pode ou não ser feito na competição. Uma destas regras define que os integrantes das duplas **não** podem ter altura somada maior que 330 cm nem menor que 300 cm. Devido ao enorme volume de inscrições nesta edição do campeonato, a organização o contratou para escrever um programa o qual, dado as alturas dos participantes, deve indicar se a dupla é válida ou não.

Entrada

A entrada é composta de uma única linha contendo dois inteiros, A e B ($125 \leq A, B \leq 220$), indicando a alturas do primeiro e do segundo integrante da dupla, em centímetros, respectivamente.

Saída

Imprima uma única linha contendo y se a dupla for válida, caso contrário, imprima uma única linha contendo n .

Exemplo de entrada 135 155	Exemplo de saída n
Exemplo de entrada 180 140	Exemplo de saída y
Exemplo de entrada 200 180	Exemplo de saída n

³Não confundir com rinha de galo.